

## REMARKS

Claims 2-20, 28-33, and 35-41 remain in the application.

### Claim Rejections under USC § 103(a)

Claims 2-5, 8-11, 14-16, 19, and 20 are rejected as being unpatentable over “Sams Teach Yourself Visual Basic 6 in 21 Days” by Greg Perry, Sams, 1998 (hereinafter Perry) in view of U.S. Patent No. 5,355,493 (“Silberbauer”) and further in view of U.S. Patent No. 5,428,554 (“Laskoski”).

Claims 6 and 7 are rejected as being unpatentable over Perry in view of Silberbauer and further in view of Laskoski and U.S. Patent No. 6,226,783 (“Limondin et al.”).

Claim 12 is rejected as being unpatentable over Perry in view of Silberbauer and further in view of Laskoski and “Understanding JavaServer Pages Model 2 Architecture”, by Govind Seshadri, December 1999 (hereinafter Seshadri).

Claim 13 is rejected as being unpatentable over Perry in view of Silberbauer and further in view of Laskoski and “What is ASP?”, by Dave Beauchemin, 4/12/00 (hereinafter Beauchemin).

Claim 17 is rejected as being unpatentable over Perry in view of Silberbauer and further in view of U.S. Patent 5,572,617 (“Eisenberg”).

Claim 18 is rejected as being unpatentable over in view of Silberbauer and further in view of U.S. Patent No. 4,992,971 (“Hayashi”).

Claim 8 is directed to a “method in a computer system for executing a program”, including “providing a plurality of definitions of interactions of the program, an interaction having one or more command definitions defining commands and a view definition, a command having input attributes and output attributes and a behavior, each interaction being associated with a request”. The method of claim 8 further includes “receiving a request”, “identifying the interaction associated with the received request” and for each command of the identified interaction, “preparing the command by setting values of input attributes of the command based on attribute values stored in an attribute store”, “performing the behavior of the command with

the input attributes” and “extracting attribute values of the command by retrieving values of the output attributes of the command and storing the retrieved values of the attributes in the attribute store”. The method of claim 8 also includes “providing a response generated by the view associated with the identified interaction”. Claim 8 further recites that “the view definition defines a view with attributes and a behavior”. In addition, claim 8 recites “setting values of input attributes of the view based on attribute values stored in an attribute store” and “performing the behavior of the view with the set attributes”.

Applicants believe that the crux of the difference in views between applicants and the Examiner in regard to the rejection of claim 8 lies in whether there is a suggestion in the prior art to support the Examiner’s proposal to combine the Laskoski reference with the Perry and Silberbauer references. In responding to applicants’ arguments submitted in January, the Examiner apparently acknowledged that the overall function of Laskoski’s software, which is to graphically display calls between functions of a computer program to aid in analysis of the program, has nothing to do with the functions of the Perry and Silberbauer references. Nevertheless, the Examiner stated that Laskoski was not relied upon for its overall function, but rather for its feature of a display function “that takes input to perform a displaying functionality and how it associates with a view definition with attributes.” Applicants respectfully submit that this explanation leaves open the question of how Laskoski or the other references provides a motivation that would lead one of ordinary skill in the art to make the combination that the Examiner proposes.

The sole rationale presented by the Examiner to justify the combination of Laskoski with Perry and Silberbauer is to “[allow] output of a command to be displayed on a computer screen so that a user can view output data.” However, this rationale is not based on any suggestion that is present in the prior art. Laskoski is only concerned with graphically representing calls between functions of a computer program, and does not teach or suggest displaying output of a command. Furthermore, neither Perry nor Silberbauer provides any suggestion to display the output of a command. Accordingly, the Examiner has failed to present a proper basis for the proposed combination of Laskoski with Perry and Silberbauer. Rather, it appears that the Examiner has impermissibly selected features one-by-one from the references to provide a

hindsight reconstruction of the invention recited in claim 8. It is therefore respectfully requested that the rejection of claim 8 be reconsidered and withdrawn.

Claims 2-7 and 9-20 are now dependent on claim 8, and are believed to be patentable on the same basis as claim 8.

In addition, claim 17 recites the limitation that “a command definition indicates whether the behavior of the command should be performed when an error is detected while performing the behavior of the commands of the interaction”.

In responding to applicants’ arguments for the separate patentability of claim 17, the Examiner has now made the following assertions:

However, Eisenberg teaches providing a [sic] safety for applications that could potentially error [sic], where the application software in Perry and Silberbauer can benefit from this safety system. Furthermore, Eisenberg teaches a rule set for a particular application software (Column 2, lines 1-5) and thus the rules are built into a command definition (Figure 1, item 12).

Applicants respectfully submit that these counter-arguments are flawed and do not provide adequate support for the Examiner’s position.

The substance of Eisenberg’s teaching is to provide safety rules for software-controlled equipment such as medical treatment or diagnostic equipment in which a software malfunction or other error could endanger the safety of a patient. It does not follow that Eisenberg teaches application of such rules to every software-controlled device in which an error could occur. The teachings of Perry and Silberbauer do not appear to be related in any way to software which controls a device in which a malfunction could endanger an individual. It is therefore respectfully submitted that there is no support in these references or in Eisenberg for the Examiner’s assertion that “the application software in Perry and Silberbauer can benefit from this safety system”. Accordingly, it appears to applicants that the Examiner has still failed to show how the prior art provides any suggestion to support the Examiner’s proposal to combine Eisenberg with the Perry and Silberbauer references.

Moreover, the passages of Eisenberg now cited by the Examiner refer to a set of safety rules, but fail to teach, as asserted by the Examiner, that “the rules are built into a command definition”. The reference is silent as to any “command definition”.

It is therefore respectfully urged that claim 17 is patentable independent of the patentability of its parent claim 8.

Claim 28 is rejected as being unpatentable over U.S. Patent No. 6,564,325 ("Travostino") in view of U.S. Patent No. 5,257,363 ("Shapiro").

Claim 29 is rejected as being unpatentable over Travostino in view of Shapiro and further in view of U.S. Patent No. 6,012,149 ("Stavran").

Claim 30 is rejected as being unpatentable over Travostino in view of Shapiro and further in view of U.S. Patent No. 5,535,390 ("Hildebrandt").

Claim 31 is rejected as being unpatentable over Travostino in view of Shapiro and further in view of Hildebrandt and U.S. Patent No. 6,038,565 ("Nock").

Claim 32 is rejected as being unpatentable over Travostino in view of Shapiro and further in view of U.S. Patent No. 5,960,441 ("Bland et al.").

Claim 28 is directed to a "method in a computer system for generating source code for a program" including "receiving a list of names of functions to be invoked by the program" and, for each of the functions to be invoked of the program, "identifying names of formal parameters of the function" and "outputting an invocation of the function that includes names of actual parameters derived from the identified names of formal parameters whereby the output invocations of the functions form the source code for the program".

In responding to applicants' argument in support of the patentability of claim 28, the Examiner appears to acknowledge that the Shapiro reference fails to teach identifying names of formal parameters or outputting an invocation of a function that includes names of actual parameters. The Examiner, however, then asserts "that parameter names and parameters themselves are interchangeable, since names represent placeholders or pointers to the actual parameters".

Applicants respectfully traverse this assertion by the Examiner. It is noted that a program will operate quite differently depending on whether parameters themselves or the names of the parameters are included in a function. Applicants therefore respectfully contend that parameter names and the parameters themselves are not "interchangeable", and that the combination of references cited by the Examiner in regard to claim 28 fails to teach or suggest the claimed invention.

It is respectfully requested that the rejection of claim 28 be reconsidered and withdrawn.

Claims 29-32 are dependent on claim 28 and are submitted as patentable on the same basis as claim 28.

Claims 33, 35 and 40 are rejected as being unpatentable over Limondin in view of U.S. Patent No. 6,032,198 ("Fujii et al.") and further in view of U.S. Patent No. 5,519,855 ("Neeman et al.") and U.S. Patent No. 6,243,862 ("Lebow").

Claims 36-38 are rejected as being unpatentable over Limondin in view of Fujii and further in view of Neeman, Lebow and Perry.

Claim 39 is rejected as being unpatentable over Limondin in view of Fujii and further in view of Neeman, Lebow and Bland.

Claim 41 is rejected as being unpatentable over Limondin in view of Fujii and further in view of Neeman, Lebow and Eisenberg.

Claim 33 is directed to a "computer-readable medium containing a data structure defining a command-based program". The data structure includes, for commands of the program, "a descriptor for the command, the descriptor identifying an object class, the object class defining input and output attributes and having a perform method". Claim 33 further recites "whereby the program is executed by instantiating objects corresponding to the object classes of the commands, setting input attribute values of objects based on output attribute values of objects, and performing the perform method of the objects". In addition, as now amended, claim 33 recites that "the commands are organized into interactions, one of the interactions including a first object class to retrieve a current application context of the application, a second object class to begin a transaction for an asset catalog, a third object class to identify a value of attributes of an asset to be added to the asset catalog, a fourth object class to store an entry identified by a created object in the asset catalog, and a fifth object class to indicate that the transaction is ending".

In explaining the rejection of claim 33, the Examiner relied upon the Fujii reference as allegedly teaching "a function for retrieving an application context of the application", and in this respect the Examiner cited a passage at column 5, lines 6-11 of the Fujii reference.

Applicants have reviewed the passage in Fujii cited by the Examiner, and it appears to applicants that that passage does not support the Examiner's reliance thereon. The passage fails to teach a function for retrieving an application context. Rather, the cited passage of Fujii refers

to FIG. 5 of the reference and discloses generating and displaying a diagram which includes a symbol representative of an application, program interface definition information, and an arrow symbol that represents a call relation between applications. This is different from retrieving an application context.

Applicants also respectfully contend that the Examiner's reliance on the Neeman reference is misplaced. The Examiner cited column 11, lines 3-4 of Neeman as allegedly teaching "identifying asset attributes". However, this passage actually refers to obtaining "property information" from a group of files. The properties referred to are properties of data files, and have nothing to do with identifying asset attributes, since the data files are not assets.

The Examiner also cited column 11, lines 5-10 of Neeman as allegedly teaching "storing an entry in a catalog". However, as is apparent from the abstract in Neeman, the catalogs referred to in that reference store query results, and thus are totally different from the asset catalogs that are the subject of claim 33.

Furthermore, the Examiner has generally failed to present any justification for the proposed combination of the Neeman reference with the other references. In this regard, applicants also note that Neeman, unlike Lebow, has nothing to do with entering data in a database, which is the subject of claim 33.

For each of these above reasons, applicants respectfully submit that the rejection of claim 33 is flawed and should be reconsidered and withdrawn.

Claims 35-41 are dependent on claim 33 and are submitted as patentable on the same basis as claim 33.

## **C O N C L U S I O N**

Accordingly, Applicants respectfully request allowance of the pending claims. If any issues remain, or if the Examiner has any further suggestions for expediting allowance of the

present application, the Examiner is kindly invited to contact the undersigned via telephone at  
(203) 972-3460.

Respectfully submitted,



June 1, 2004

Date

Nathaniel Levin  
Registration No. 34,860  
Buckley, Maschoff & Talwalkar LLC  
Five Elm Street  
New Canaan, CT 06840  
(203) 972-3460